*# EXAMPLES*

*### nice representation (1)*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), categ.class.order = list(NULL, c("B", "A")), categ.legend.name = "LEGEND", categ.color = NULL, box.width = 0.3, whisker.width = 0.8, dot.color = "same", dot.jitter = 0.5, dot.size = 3.5, dot.border.size = 0.2, dot.alpha = 0.5, ylim = c(10, 25), y.include.zero = TRUE, stat.disp = "above", stat.size = 4, xlab = "GROUP", ylab = "VALUE", text.size = 12, title = "GRAPH1", title.text.size = 8, text.angle = 0, classic = TRUE, grid = TRUE)*

*### nice representation (2)*

*set.seed(1) ; obs1 <- data.frame(Time = c(rnorm(24, 0), rnorm(24, -10), rnorm(24, 10), rnorm(24, 20)), Group1 = rep(c("CAT", "DOG"), times = 48), Group2 = rep(c("A", "B", "C", "D"), each = 24)) ; set.seed(NULL) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), categ.class.order = list(NULL, c("B", "A", "D", "C")), categ.legend.name = "LEGEND", categ.color = NULL, box.width = 0.8, dot.color = "grey50", dot.tidy = TRUE, dot.bin.nb = 60, dot.size = 3.5, dot.border.size = 0.2, dot.alpha = 0.5, ylim= c(-20, 30), stat.disp = "above", stat.size = 4, stat.dist = 1, xlab = "GROUP", ylab = "VALUE", vertical = FALSE, text.size = 12, title = "GRAPH1", title.text.size = 8, text.angle = 45, classic = FALSE)*

*### separate boxes. Simple example*

*set.seed(1) ; obs1 <- data.frame(Time = c(rnorm(10), rnorm(10) + 2), Group1 = rep(c("G", "H"), each = 10)) ;*

*fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1")*

*### separate boxes. Changing the order of the boxes*

*fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", categ.class.order = list(c("H", "G")))*

*### separate boxs. Example (1) of modification of box color using a single value*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", categ.color = "white")*

*### separate boxs. Example (2) of modification of box color using one value par class of categ2*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", categ.color = c("coral", "lightblue"))*

*### separate boxs. Example (3) of modification of box color using the box.color data frame column, with respect of the correspondence between categ2 and box.color columns*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), box.color = rep(c("coral", "lightblue"), time = 10)) ; obs1 ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", categ.color = obs1$box.color)*

*### separate boxs. Example (1) of modification of dot color, using the same dot color as the corresponding box*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", dot.color = "same")*

*### separate boxs. Example (2) of modification of dot color, using a single color for all the dots*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", dot.color = "green")*

*### separate boxs. Example (3) of modification of dot color, using one value par class of categ2*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", dot.color = c("green", "brown"))*

*### separate boxs. Example (4) of modification of dot color, using different colors for each dot*

*obs1 <- data.frame(Time = 1:10, Group1 = rep(c("G", "H"), times = 5)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", dot.color = hsv(h = (1:nrow(obs1)) / nrow(obs1)))*

*### grouped boxs. Simple example*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"))*

*### grouped boxs. More grouped boxs*

*obs1 <- data.frame(Time = 1:24, Group1 = rep(c("G", "H"), times = 12), Group2 = rep(c("A", "B", "C", "D"), each = 6)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"))*

*### grouped boxs. Example (1) of modification of box color, using a single value*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), categ.color = "white")*

*### grouped boxs. Example (2) of modification of box color, using one value par class of categ2*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), categ.color = c("coral", "lightblue"))*

*### grouped boxs. Example (3) of modification of box color, using one value per line of obs1, with respect of the correspondence between categ2 and box.color columns*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10), box.color = rep(c("coral", "lightblue"), each = 10)) ; obs1 ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), categ.color = obs1$box.color)*

*### grouped boxs. Example (1) of modification of dot color, using the same dot color as the corresponding box*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = "same")*

*### grouped boxs. Example (2) of modification of dot color, using a single color for all the dots*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = "green")*

*### grouped boxs. Example (3) of modification of dot color, using one value par class of categ2*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = c("green", "brown"))*

*### grouped boxs. Example (4) of modification of dot color, using different colors for each dot*

*obs1 <- data.frame(Time = 1:10, Group1 = rep(c("G", "H"), times = 5), Group2 = rep(c("A", "B"), each = 5)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = hsv(h = (1:nrow(obs1)) / nrow(obs1)))*

*### no dots (y.include.zero set to TRUE to see the lowest box):*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = NULL, y.include.zero = TRUE)*

*### box width. Example (1) with box.width = 0.25 -> three times more space between single boxs than the box width (y.include.zero set to TRUE to see the lowest box)*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), each = 500)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", dot.color = NULL, y.include.zero = TRUE, box.width = 0.25)*

*### box width. Example (2) with box.width = 1, no space between single boxs*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), each = 500)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = "Group1", dot.color = NULL, y.include.zero = TRUE, box.width = 1)*

*### box width. Example (3) with box.width = 0.25 -> three times more space between sets of grouped boxs than the set width*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), times = 500), Group2 = rep(LETTERS[1:5], each = 200)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = NULL, y.include.zero = TRUE, box.width = 0.25)*

*### box width. Example (4) with box.width = 0 -> no space between sets of grouped boxs*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), times = 500), Group2 = rep(LETTERS[1:5], each = 200)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = NULL, y.include.zero = TRUE, box.width = 1)*

*### whisker width. Example (1) with whisker.width = 1 -> whiskers have the width of the corresponding box*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), times = 500), Group2 = rep(LETTERS[1:5], each = 200)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = NULL, whisker.width = 1)*

*### whisker width. Example (2) error boxs with no whiskers*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), times = 500), Group2 = rep(LETTERS[1:5], each = 200)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = NULL, whisker.width = 0)*

*### tidy dot distribution. Example (1)*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), times = 500), Group2 = rep(LETTERS[1:5], each = 200)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = "same", dot.tidy = TRUE, dot.bin.nb = 100)*

*### tidy dot distribution. Example (2) reducing the dot size with dot.bin.nb*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), times = 500), Group2 = rep(LETTERS[1:5], each = 200)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = "same", dot.tidy = TRUE, dot.bin.nb = 150)*

*### dot jitter. Example (1)*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), times = 500), Group2 = rep(LETTERS[1:5], each = 200)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = "same", dot.tidy = FALSE, dot.jitter = 1, dot.size = 2)*

*### dot jitter. Example (2) with dot.jitter = 1 -> dispersion around the corresponding box width*

*obs1 <- data.frame(Time = 1:1000, Group1 = rep(c("G", "H"), times = 500), Group2 = rep(LETTERS[1:5], each = 200)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = "grey", dot.size = 3, dot.alpha = 1, dot.jitter = 1)*

*### dot jitter. Example (3) with no dispersion*

*obs1 <- data.frame(Time = 1:100, Group1 = rep(c("G", "H"), times = 50), Group2 = rep(LETTERS[1:5], each = 20)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = "grey", dot.size = 3, dot.alpha = 1, dot.jitter = 0)*

*### dot size, dot border size and dot transparency*

*obs1 <- data.frame(Time = 1:100, Group1 = rep(c("G", "H"), times = 50), Group2 = rep(LETTERS[1:5], each = 20)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), dot.color = "grey", dot.size = 4, dot.border.size = 0, dot.alpha = 0.6)*

*### y-axis limits. Example (1)*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylim = c(-1, 25))*

*### y-axis limits. Example (2) showing that order matters in ylim argument*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylim = c(25, -1))*

*### log scale. Example (1). BEWARE: y column must be log, otherwise incoherent scale (see below warning message with the return argument)*

*obs1 <- data.frame(Time = log10((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylog = "log10")*

*### log scale. Example (2). BEWARE: values of the ylim must be in the corresponding log*

*obs1 <- data.frame(Time = log10((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylog = "log10", ylim = c(1,4))*

*### tick number. Example (1)*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), y.tick.nb = 10)*

*### tick number. Example (2) using a log2 scale*

*obs1 <- data.frame(Time = log2((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylog = "log2", y.tick.nb = 10, ylim = c(1, 16))*

*### tick number. Example (3) using a log10 scale*

*obs1 <- data.frame(Time = log10((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylog = "log10", y.tick.nb = 10)*

*### tick number. Example (4) using a log10 scale: the reverse y-axis correctly deal with log10 scale*

*obs1 <- data.frame(Time = log10((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylog = "log10", y.tick.nb = 10, ylim = c(4, 1))*

*### secondary tick number. Example (1)*

*obs1 <- data.frame(Time = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), y.inter.tick.nb = 2)*

*### secondary ticks. Example (2) not for log2 and log10 scales (see below warning message with the return argument)*

*obs1 <- data.frame(Time = log10((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylog = "log10", y.inter.tick.nb = 2)*

*### include zero in the y-axis*

*obs1 <- data.frame(Time = (1:20), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), y.include.zero = TRUE)*

*### extra margins. To avoid dot cuts*

*obs1 <- data.frame(Time = (1:20), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), y.top.extra.margin = 0.25, y.bottom.extra.margin = 0.25)*

*### mean diplay. Example (1) at the top of the plot region*

*obs1 <- data.frame(Time = (1:20), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), y.top.extra.margin = 0.1, stat.disp = "top", stat.size = 4, stat.dist = 2)*

*### mean diplay. Example (2) above boxs*

*obs1 <- data.frame(Time = (1:20), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), y.top.extra.margin = 0.1, stat.disp = "above", stat.size = 4, stat.dist = 2)*

*### box orientation. Example (1) without log scale, showing that the other arguments are still operational*

*obs1 <- data.frame(Time = (1:20), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), y.tick.nb = 10, y.inter.tick.nb = 2, y.include.zero = TRUE, vertical = FALSE)*

*### box orientation. Example (2) with log scale. Horizontal orientation is blocked with log2 and log10 scales because of a bug in ggplot2 (https://github.com/tidyverse/ggplot2/issues/881)*

*obs1 <- data.frame(Time = log10((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylog = "log10", vertical = FALSE)*

*### classic representation (use grid = TRUE to display the background lines of the y axis ticks)*

*obs1 <- data.frame(Time = (1:20), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), classic = TRUE, grid = FALSE)*

*### graphic info. Example (1)*

*obs1 <- data.frame(Time = log10((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), return = TRUE)*

*### graphic info. Example (2) of assignation and warning message display*

*obs1 <- data.frame(Time = log10((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; warn <- fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), ylog = "log10", return = TRUE) ; cat(warn$warnings)*

*### add ggplot2 functions*

*obs1 <- data.frame(Time = log10((1:20) \* 100), Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "Time", categ = c("Group1", "Group2"), add = "+ggplot2::theme\_classic()")*

*### all the arguments*

*obs1 <- data.frame(x = 1:20, Group1 = rep(c("G", "H"), times = 10), Group2 = rep(c("A", "B"), each = 10)) ; fun\_gg\_boxplot(data1 = obs1, y = "x", categ = c("Group1", "Group2"), categ.class.order = list(NULL, c("B", "A")), categ.legend.name = "", categ.color = c("red", "blue"), box.width = 0.25, whisker.width = 0.8, dot.color = "grey", dot.tidy = FALSE, dot.bin.nb = 30, dot.jitter = 1, dot.size = 4, dot.border.size = 0, dot.alpha = 1, ylim = c(0, 25), ylog = "no", y.tick.nb = NULL, y.inter.tick.nb = NULL, y.include.zero = FALSE, y.top.extra.margin = 0.05, y.bottom.extra.margin = 0, stat.disp = "above", stat.size = 4, stat.dist = 2, xlab = "GROUP", ylab = "VALUE", vertical = FALSE, text.size = 12, title = "", title.text.size = 8, text.angle = 45, classic = TRUE, grid = TRUE, return = TRUE, plot = TRUE, add = NULL, warn.print = TRUE, path.lib = NULL)*

*# problem of warning message*

*# error with dot.tidy = TRUE*